NAT'L INST. OF STAND & TECH
A11106 494875
NIST PUBLICATIONS
REFERENCE

NISTIR 7228

Boundary/Contour Fitted Grid Generation for Effective Visualizations in a Digital Library of Mathematical Functions

Bonita Saunders
Qiming Wang

QC
100
.U56
#7228
2005

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7228

Boundary/Contour Fitted Grid Generation for Effective Visualizations in a Digital Library of Mathematical Functions

Bonita Saunders
Mathematical and
Computational Sciences Division
Information Technology Laboratory

Qiming Wang
Information Access Division
Information Technology Laboratory

August 2005

U.S. DEPARTMENT OF COMMERCE
Carlos M. Gutierrez, Secretary
TECHNOLOGY ADMINISTRATION
Michelle O'Neill, Acting Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
William Jeffrey, Director

Boundary/Contour Fitted Grid Generation for Effective Visualizations in a Digital Library of Mathematical Functions

Bonita Saunders
Qiming Wang

National Institute of Standards and Technology
100 Bureau Drive Stop 8910
Gaithersburg, MD 20899-8910 USA
bonita.saunders@nist.gov, qiming.wang@nist.gov

Abstract

Effective visualizations can help researchers obtain a more complete understanding of high level mathematical functions that arise in mathematics, statistics, physics, fluid dynamics and other fields of the mathematical and physical sciences. Accordingly, dynamic interactive 3D graphs of function surfaces will be a key feature of the National Institute of Standards and Technology Digital Library of Mathematical Functions, a new Web-based compendium of mathematical functions that will replace a popular but dated resource, the National Bureau of Standards Handbook of Mathematical Functions, published by Abramowitz and Stegun in 1964. As developers of commercial packages are well aware, creating software to accurately plot complicated 3D surfaces can be a challenging task. This paper looks at the effectiveness of modifying an algebraic tensor product spline grid generation technique, whose design was originally motivated by problems in aerodynamics and solidification theory, to create computational grids for accurate visualizations of 3D surfaces that capture key function features such as poles, branch cuts and other singularities.

Introduction

The National Institute of Standards and Technology (NIST) is in the midst of developing the NIST Digital Library of Mathematical Functions (DLMF) to replace the widely used, but outdated National Bureau of Standards Handbook of Mathematical Functions published in 1964 [1]. Like the original book, the digital library will contain a wide range of information about high level, or special, mathematical functions such as the Bessel functions, the gamma and beta functions, hypergeometric functions and others useful for obtaining closed form solutions or qualitative information to solve many problems in the mathematical and physical sciences. There will be formulas, methods of computation, references, and links to software for over forty functions. The Web site will feature interactive navigation, a mathematical equation search, 2D graphics, and dynamic interactive 3D visualizations that allow users to examine poles, zeros, branch cuts and other important features of special functions.

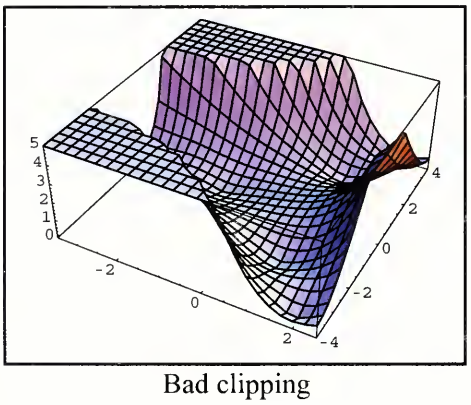
This paper discusses our successful use of numerical grid generation techniques to facilitate the plotting of surfaces that are the graphs of high level mathematical functions, that is, surfaces that can be described by equations of the form $w=f(x,y)$. The complicated nature of these functions means that the computational domains are often irregular, discontinuous, or multiply connected, making the visualization task quite difficult. Although commercial packages may have some of the functions built in and may allow the user to produce a plot, typically computations are over a rectangular Cartesian mesh, which sometimes produces a very poor and misleading graph of the function. In addition, the packages often have problems clipping the surface properly when values fall outside the range of interest specified by the user. Furthermore, what looks satisfactory inside the package, may not be when the data is transformed to a format more suitable for the Web.

Many problems can be eliminated by designing a computational grid whose boundary coincides with a selected contour of the surface. This can not only produce a proper clipping of the function, but also improve the smoothness of the color mapping. Various structured and unstructured techniques might be used to create the computational grids, but structured techniques make it easier to write efficient code to implement the interactive capabilities for the visualizations. We will first examine the problems that can arise when trying to display dynamic 3D graphics on the Web. Then we will discuss the grid generation technique we have used, examining an updated version of a tensor product B-spline grid generation algorithm designed by one of the authors [2]. Finally, we will discuss our results and look at some possibilities for improvements.

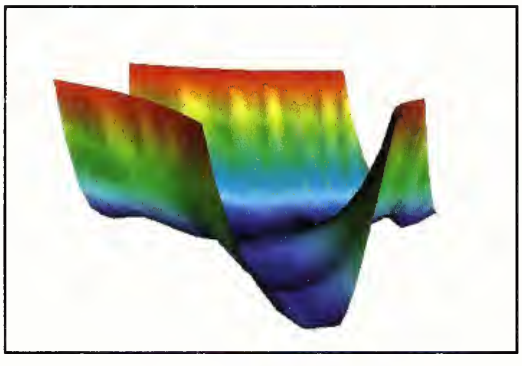
Problem

The content of the NIST DLMF will be organized into approximately forty chapters written by renowned mathematicians throughout the United States and abroad. The locations and types of graphs and visualizations for a particular chapter are determined by consultation with the DLMF editors and chapter authors. Whenever possible, data accuracy is being validated through computation by at least two different methods using commercial packages, published software, or the author's personal codes. In addition to these issues, a key concern is plot accuracy. Is the displayed plot an accurate representation of the graph of the function?

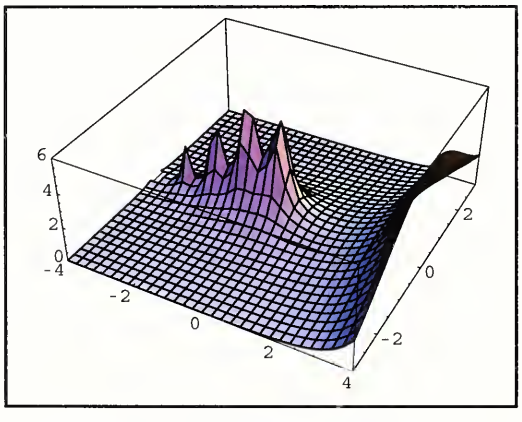
Most commercial packages handle 2D graphics very well. If the user only wants to see the function values that lie within a particular interval, they properly cut, or clip, the curve so that only points falling within the desired range appear. Discontinuities are usually handled either automatically or fairly easily with special options. Unfortunately, the situation is often different with 3D graphics. Figure 1 shows some of the problems we have encountered when using commercial packages. In the first picture the user has requested that Airy function $|\mathrm{Bi}(z)|$ be plotted only for values less than or equal to 5. The package produces what might be called a shelf, or table effect, where function values that exceed the value 5 are set to 5. Although the flat area, which is totally unrelated to the function, might not concern some users, it might be confusing to students and others unfamiliar with the function. The figure on the right shows that even when clipping is done properly, an undesirable color map may result when the data is computed over purely rectangular mesh cells and rendered in a format for Web viewing. The surface was displayed using the Virtual Reality Modeling Language (VRML), a standard 3D file format for creating interactive Web-based visualizations [4]. The bottom figure, showing the complex gamma function, illustrates the lack of clarity packages may demonstrate when functions have complicated features such as poles.



Bad clipping



Non-smooth color map



Poor resolution of poles

Figure 1. Potential problems.

Most of these problems can be eliminated or decreased in severity by computing the special function over a boundary or contour fitted grid. The next section discusses the grid generation mapping we have used to produce such a grid.

The Grid Generation Mapping

The difficulty of the grid generation problem depends directly on the complexity of the computational domain for the special function. The shapes of the domains range from simple rectangles to oddly shaped multiply connected domains which must be split to handle branch cuts. We have experimented with both structured and unstructured grids, but we have found that structured grids allow us to write more efficient code to implement the interactive features for the visualizations. One of the primary grid generation techniques we have used is based on a tensor product mapping and smoothing functional developed by one of the authors to solve partial differential equations (pdes) related to aerodynamics and solidification theory[2,3]. The grid generation technique uses a tensor product mapping **T** from the unit square I_2 to the physical domain defined by

$$\mathbf{T}(\xi,\eta) = \begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \Sigma_{i=1}^{m}\Sigma_{j=1}^{n}\alpha_{ij}B_{ij}(\xi,\eta) \\ \Sigma_{i=1}^{m}\Sigma_{j=1}^{n}\beta_{ij}B_{ij}(\xi,\eta) \end{pmatrix}$$

where $0 \le \xi, \eta \le 1$, $B_{ij}(\xi,\eta) = B_i(\xi)B_j(\eta)$ and B_i, B_j are elements of cubic B-spline sequences. An advantage of this technique is that the coefficients α_{ij}, β_{ij} can be easily chosen to produce a very good variation diminishing approximation to transfinite interpolation if the grid is very simple[5], or for more complicated highly nonconvex boundaries, it can become a variational method with the coefficients chosen to minimize the functional

$$F = \int_{I_2} (w_1\{(\frac{\partial J}{\partial \xi})^2 + (\frac{\partial J}{\partial \eta})^2\} + w_2\{\frac{\partial \mathbf{T}}{\partial \xi}\cdot\frac{\partial \mathbf{T}}{\partial \eta}\}^2 + w_3\{Ju\})dA$$

where J is the Jacobian of the mapping, w_1, w_2, w_3 are weight constants and u represents external criteria for adapting the grid. Like the variational method of Brackbill and Saltzman[6], the integral controls smoothness, orthogonality, and when w_3 is nonzero, the adaptive concentration of the grid lines based on the definition of u. When solving pdes, u might be a function of the gradient of the evolving solution or an approximation of truncation error. Here we want u to be based on curvature and gradient information related to the function surface. Figure 2 shows the initial and optimized grids created for a puzzle shaped boundary. The minimization of the functional with $w_1 = 2, w_2 = 10, w_3 = 0$ pulls all the lines inside the boundary and smoothes the grid cell spacing.

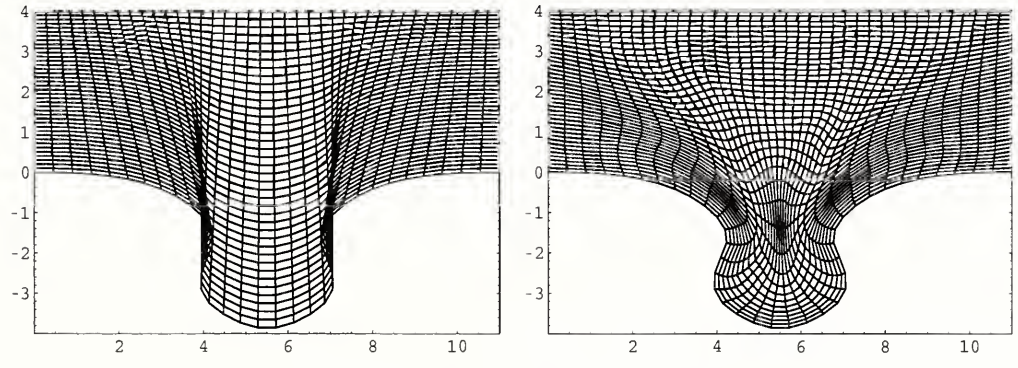


Figure 2. Initial and optimized puzzle grids.

Results

To date we have used boundary/contour fitted grid generation to create computational grids for the production of over one hundred 3D visualizations for the NIST DLMF project. The plots in Figures 3 and 4 vividly illustrate the effectiveness of the technique. In Figure 3, the left picture shows the contour curves for |Bi' (z)| = 5. We connected the curves to parts of a rectangle to form a closed boundary and created the contour fitted grid shown on the right. The left plot in Figure 4 shows the surface obtained after we computed the function over the grid and translated the data into VRML format for viewing on the Web. The figure on the right shows the same function surface computed over a rectangular Cartesian grid. Notice that the contour fitted grid not only properly clips the function, but also greatly improves the smoothness of the color map.

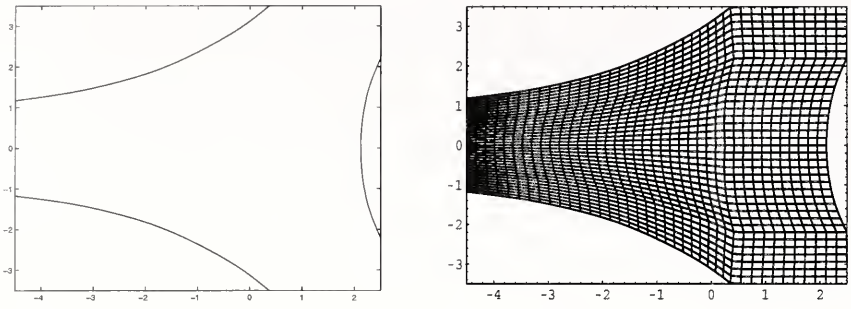


Figure 3. Contours and Grid for Airy function |Bi' (z)|.

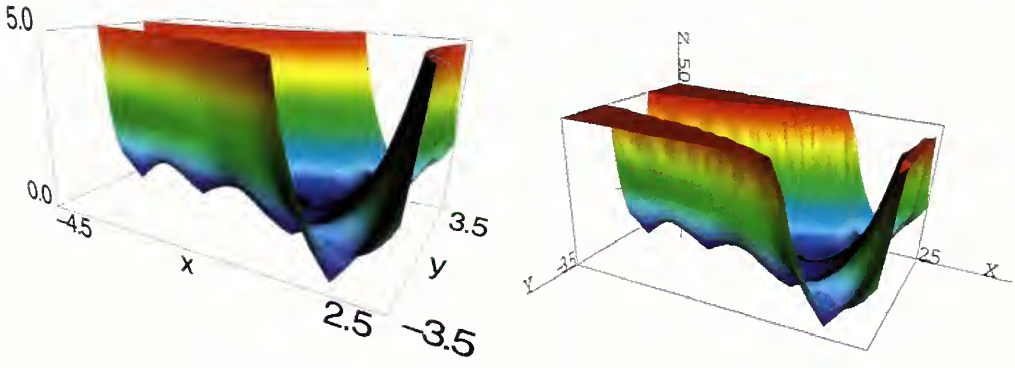


Figure 4. Airy function |Bi' (z)|.

The next three figures show grids and surfaces for several types of Weierstrass elliptic functions. All the functions have several poles, making the computational domains of the functions somewhat complicated. To ease the complexity of the grid generation problem, we use a block-structured approach, dividing the domains into several sections and creating a grid on each. Since we are not solving pdes, we are not concerned with matching boundary conditions across the separate grids. Furthermore, our VRML code, which governs interaction with the surfaces, accepts any number of grids and operates on each grid independently.

Figure 5 shows the grid for Weierstrass elliptic function $|\wp(3.7; a+ib, 0)|$ where $-5 \le a \le 3, -4 \le b \le 4$. The bottom half of the grid ($b<0$) was reflected around the line $b=0$ to obtain the top half. We used an exponential function to concentrate the grid points.

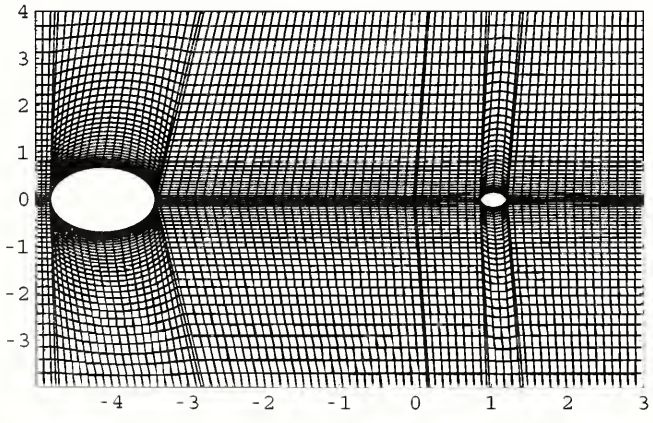


Figure 5. Grid for Weierstrass Elliptic Function $|\wp(3.7; a+ib, 0)|$.

Once the coefficients are determined for the tensor product mapping we have an algebraic method which can be used to create a coarser, or finer, grid simply by evaluating fewer, or more, points on the unit square. To guarantee that key boundary points, such as those located at function zeros, are maintained regardless of grid size, we identify "fixed points," that is, boundary points which must always be kept. Grid lines are always drawn through these points. The resulting discontinuities in cell spacing do not appear to affect the quality of the 3D visualizations. In Figure 6 we show a display of the function computed over the grid in Figure 5 inside a VRML Cosmo Player browser. Most VRML browsers have built-in capabilities for rotation, pan, and zoom, but also allow a programmer to add custom features. We have added user-controlled options for scaling, alternate color maps, and movable cutting planes.

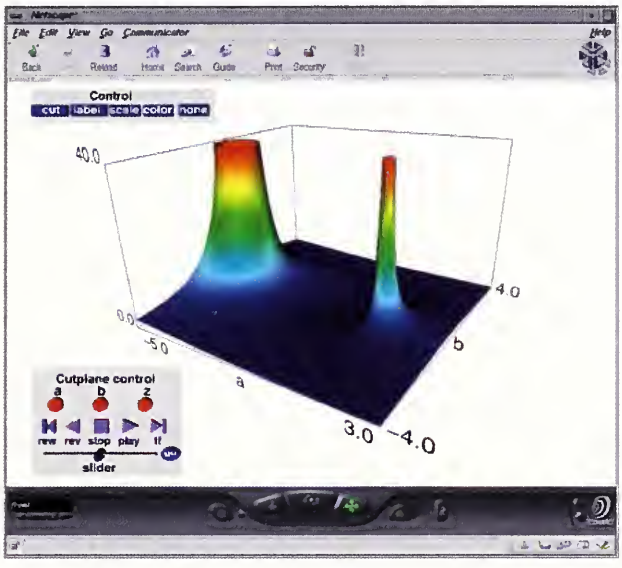


Figure 6. Weierstrass Elliptic Function $|\wp(3.7;a+ib,0)|$.

Figure 7 shows one of four sections of the grid for the Weierstrass $\wp$ – function $|\wp(x+iy;1,4i)|$, and Figure 8 shows one of two sections of the grid for the Weierstrass $\wp$ – function $|\wp(x+iy)|$. For both grids the most time consuming part of the process was computing the contours for the grid boundaries, but the symmetry of the surface in Figure 8 made the task somewhat easier. In the Weierstrass grids we have set $w_3 = 0$ so that adaptive concentration of the grid lines is not applied, but in Figure 9 we show preliminary progress we have made in this area.

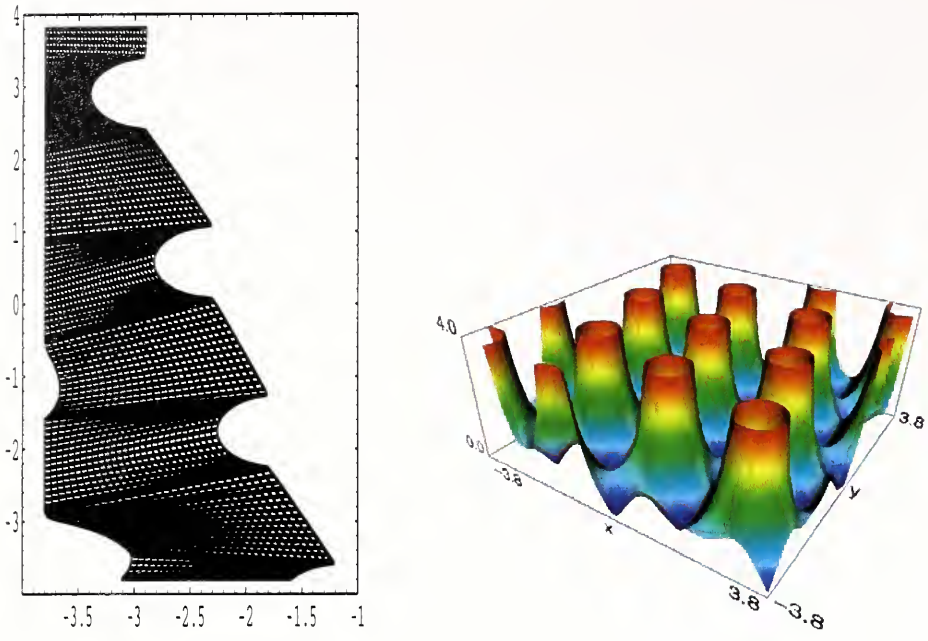


Figure 7. Weierstrass Elliptic Function $|\wp(x+iy;1,4i)|$.

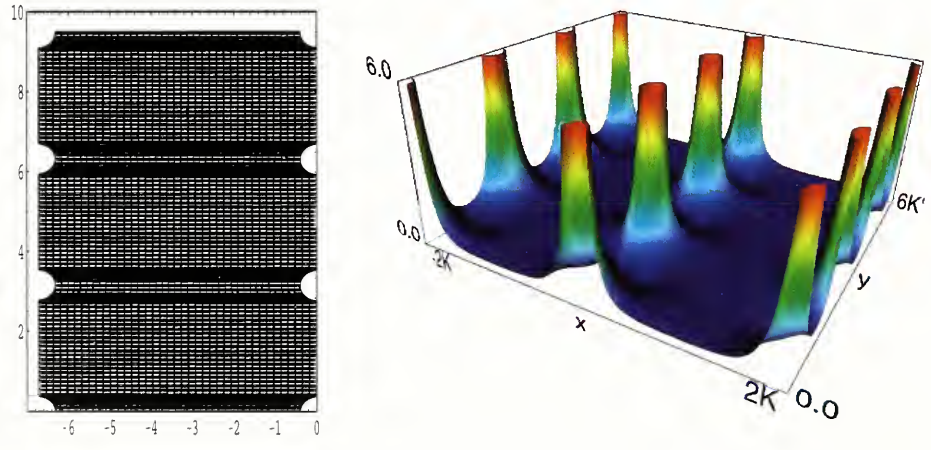


Figure 8. Weierstrass Elliptic Function $|\wp(x+iy)|$.

In Figures 7 and 8 the smoothness of the color map is satisfactory for both surfaces, but there are some shadowy areas caused by the skewness of the grids near the ends of the semicircular contours. These areas might be improved by adaptively concentrating the grid lines according to gradient and curvature information obtained from the function. To date, we have successfully attracted grid lines to several pre-defined curves. In Figure 9 we attract grid points in an equally spaced square grid to a circle by choosing Jacobian and orthogonality weights, $w_1 = 3, w_2 = 1$, respectively, and adaptive weight $w_3 = 3$.

We define u on I_2 by $u(\xi,\eta) = 5000e^{-500[(\xi-.5)^2+(\eta-.5)^2-9/64]^2}$. Currently, we are working on adaptive movement based on function curvature, but several problems must be addressed. Functional minimization routines, originally designed only for $w_3 = 0$, are being updated to accommodate a likely nonlinear dependence of u on the spline coefficients. Also, the specialized nature of many of the functions means that accessing and linking the codes needed to compute gradient and curvature data may not be a trivial task.

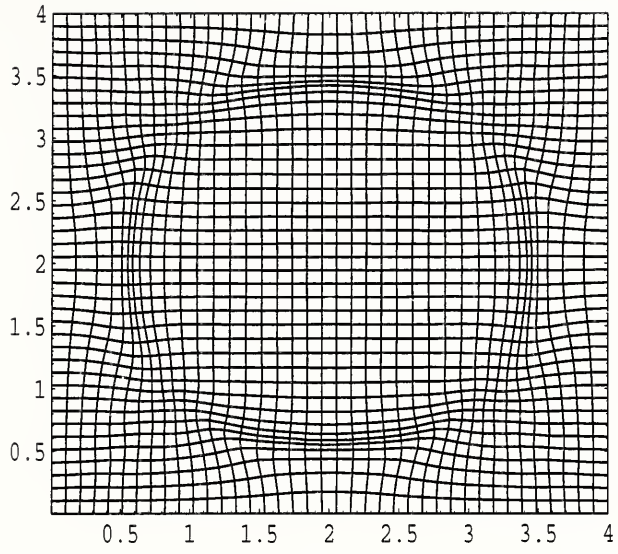


Figure 9. Grid adapted to circular shape.

Conclusions

We have successfully used boundary/contour fitted grid generation to create over one hundred 3D visualizations of complex mathematical functions for the NIST Digital Library of Mathematical Functions Project. The grid generation technique has helped us address many of the problems such as poor clipping, inaccurate plots, and poor color mapping that can come with using standard commercial packages for 3D graphics. We continue to improve the technique and have made significant progress toward implementing the adaptive movement of grid lines based on external information such as function curvature and gradient data.

Disclaimer

All references to commercial products are provided only for clarification of the results presented. Their identification does not imply recommendation or endorsement by NIST.

References

[1] Abramowitz, M and Stegun, I.A. editors, *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Vol. 55, National Bureau of Standards Applied Mathematics Series. U.S. Government Printing Office, Washington, D.C., 1964.

[2] Saunders, B.V. "A Boundary Conforming Grid Generation Stystem for Interface Tracking," *J. Computers Math Applic . 29,1-17*, 1995.

[3] Saunders, B.V., Wang, Q. "From 2D to 3D: Numerical Grid Generation and the Visualization of Complex Surfaces" *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations,* Soni, B.K., et all (Editors), Whistler, British Columbia, Canada, September 25-28, 2000.

[4] VRML. The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997.

[5] de Boor, C., *A Practical Guide to Splines*, Revised Edition, Springer-Verlag, New York, 2001.

[6] Brackbill, J.U. and Saltzman, J.S. "Adaptive Zoning for Singular Problems in Two Dimensions," *J. Comput. Phys*. 46, 342-368, 1982.